Adversarial Attack and Detection under the Fisher Information Metric

Chenxiao Zhao, P. Thomas Fletcher, Mixue Yu, Yaxin Peng, Guixu Zhang, Chaomin Shen

December 22, 2018

Outline

Outline

What do we know about adversarial examples?

- Some imperceptible noise added on the input can alter the output prediction[1]

go-kart

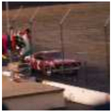
racer

ski

running shoe

amphibian

balance beam



alp

sandal

What do we know about adversarial examples?

- Some imperceptible noise added on the input can alter the output prediction[1]
- Transfer between different models[2]

[1] I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". In: *ArXiv preprints arXiv:1412.6572* (2014).

[2] C. Szegedy et al. "Intriguing properties of neural networks". In: *ArXiv preprints arXiv:1312.6199* (2013).

[3] F. Tramèr et al. "The space of transferable adversarial examples". In: *arXiv preprint arXiv:1704.03453* (2017).

What do we know about adversarial examples?

- Some imperceptible noise added on the input can alter the output prediction[1]
- Transfer between different models[2]
- Generally exist in a large and continuous subspace[3]

[1] I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". In: *ArXiv preprints arXiv:1412.6572* (2014).

[2] C. Szegedy et al. "Intriguing properties of neural networks". In: *ArXiv preprints arXiv:1312.6199* (2013).

[3] F. Tramèr et al. "The space of transferable adversarial examples". In: *arXiv preprint arXiv:1704.03453* (2017).

Characterizing the vulnerability of deep learning models

How to characterize the vulnerability of a deep learning model?

- Worst case perturbation[4]
- Satisfiability modulo theory (SMT) solver[5]
- loss surface / local curvature

[4] A. Sinha, H. Namkoong, and J. Duchi. "Certifying some distributional robustness with principled adversarial training". In: *ArXiv preprints arXiv:1710.10571* (2017).

[5] G. Katz et al. "Reluplex: An efficient SMT solver for verifying deep neural networks". In: *International Conference on Computer-Aided Verification*. 2017, pp. 97–117.

Characterizing the vulnerability of deep learning models

How to characterize the vulnerability of a deep learning model?

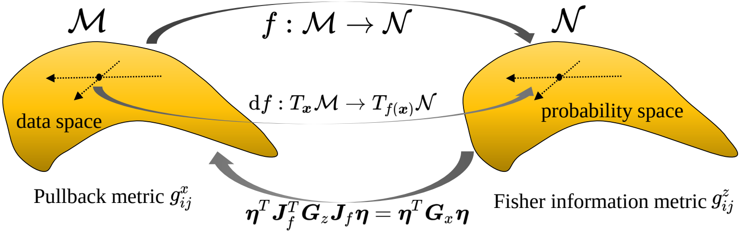
- Worst case perturbation[4]
- Satisfiability modulo theory (SMT) solver[5]
- loss surface / local curvature

In general, the previous approaches regard the neural network as a function mapping $f : \mathbb{R}^m \to \mathbb{R}^n$.

[4] A. Sinha, H. Namkoong, and J. Duchi. "Certifying some distributional robustness with principled adversarial training". In: *ArXiv preprints arXiv:1710.10571* (2017).

[5] G. Katz et al. "Reluplex: An efficient SMT solver for verifying deep neural networks". In: *International Conference on Computer-Aided Verification*. 2017, pp. 97–117.

The Fisher information metric approach



Pullback metric

Definition (pullback metric)

Let $\phi : \mathcal{M} \to \mathcal{N}$ is a differentiable map, and $\mathcal{N}$ is a Riemannian manifold with metric tensor $g^{\mathcal{N}}$, then the pullback of $g^{\mathcal{N}}$ along ϕ is a quadratic form on the tangent space of $\mathcal{M}$. Given $p \in \mathcal{M}$ and $v, w \in T_p\mathcal{M}$, the quadratic form $g^{\mathcal{M}}$ is given by

$$g^{\mathcal{M}}(v, w) = g^{\mathcal{N}}(d\phi(v), d\phi(w))$$

where $d\phi(v) : T_v\mathcal{M} \to T_{\phi(v)}\mathcal{N}$ is the pushforward of v by ϕ.

Objective function

For adversarial attacks, the goal is to find a subtle perturbation η for a given input, such that the output prediction varies from the the correct to the wrong output.

$$\max_{\eta} \eta^T g^x \eta \qquad \text{s.t. } \|\eta\|_2^2 = \epsilon$$

- The optimal solution for η is the first eigenvector of matrix g^x
- But how do we define the metric tensor g^x?

Fisher information

Definition (Fisher information)

Let $p(x|\theta)$ be a probability density function of random variable X conditioned on parameter θ. The Fisher information matrix of θ, denoted as g^θ, is defined as the variance of the expectation over the derivative of log-likelihood with respect to θ:

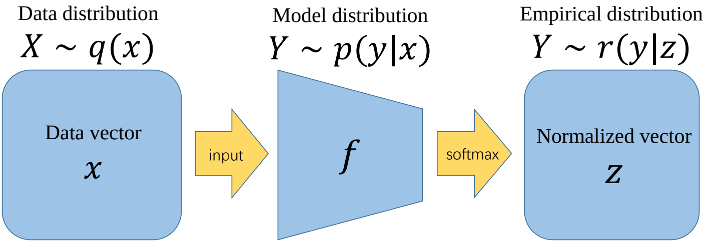
$$g_{ij}^\theta = \mathbb{E}_{x|\theta}[(\frac{\partial}{\partial\theta_i}\log p(x|\theta))(\frac{\partial}{\partial\theta_j}\log p(x|\theta))^T]$$

Many theoretical advantages in[6]

[6] S. Amari and H. Nagaoka. *Methods of Information Geometry*. Providence, RI: American Mathematical Society, 2007.

Outline

FIM of the input samples



FIM of the input samples

- For adversarial attacks, the input x is the only changeable variable. With some reparameterization of variables we obtain

$$g_{ij}^{x} = \mathbb{E}_{y|x}[(\frac{\partial}{\partial x_i}\log p(y|x))(\frac{\partial}{\partial x_j}\log p(y|x))^T]$$

[7] Hyeyoung Park, S-I Amari, and Kenji Fukumizu. "Adaptive natural gradient learning algorithms for various stochastic models". In: *Neural Networks* 13.7 (2000), pp. 755–764.

FIM of the input samples

- For adversarial attacks, the input x is the only changeable variable.
- Let J_f be the Jacobian of $f(x)$ w.r.t. x. Using the definition of pullback metric, FIM can be calculated by[7]

$$
\begin{aligned}
g^x &= J_f^T \mathbb{E}_{y|f(x)}[(\frac{\partial}{\partial z} r(y|z))(\frac{\partial}{\partial z} r(y|z))^T] J_f \\
&= J_f^T g^z J_f
\end{aligned}
$$

FIM of the input samples

- For adversarial attacks, the input x is the only changeable variable.
- Let J_f be the Jacobian of $f(x)$ w.r.t. x. Using the definition of pullback metric, FIM can be calculated by[7]

$$g^x = J_f^T g^z J_f$$

- Given η as the adversarial perturbation, a general approach is to compute the Hessian of the KL divergence[8]

$$g_{ij}^x = \mathbb{E}_{y|f(x)}[\frac{\partial^2}{\partial x_i \partial x_j} KL(p(y|x)||p(y|x+\eta))]$$

[7] Hyeyoung Park, S-I Amari, and Kenji Fukumizu. "Adaptive natural gradient learning algorithms for various stochastic models". In: *Neural Networks* 13.7 (2000), pp. 755–764.

[8] Takeru Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* (2018).

FIM of the input samples

- How can we calculate FIM more efficiently?
- How can we apply FIM to the objective functions which might not involve a probabilistic model in any obvious way?
- An solution that combines both accuracy and efficiency is to use the empirical Fisher[9]

$$g_{ij}^{x} = \mathbb{E}_{r(y|z)}[(\frac{\partial}{\partial x_i} \log p(y|x))(\frac{\partial}{\partial x_j} \log p(y|x))^T]$$

[9] James Martens. "New insights and perspectives on the natural gradient method". In: *arXiv preprint arXiv:1412.1193* (2014).

Why empirical Fisher?

What are the advantages for using the empirical distribution instead of true underlying distribution?

- The empirical Fisher is essentially easy to compute, provided that one is already calculating the gradient

$$g^x = \sum_{i=1}^{n} r_i(y|z)[(\frac{\partial}{\partial x} \log p_i(y|x))(\frac{\partial}{\partial x} \log p_i(y|x))^T]$$

Why empirical Fisher?

What are the advantages for using the empirical distribution instead of true underlying distribution?

- The empirical Fisher is essentially easy to compute, provided that one is already calculating the gradient
- $\mathrm{rank}(g^x) \leq \mathrm{rank}(g^z) = m$, making optimization strategies for sparse matrices applicable (Lanczos method)

Why empirical Fisher?

What are the advantages for using the empirical distribution instead of true underlying distribution?

- The empirical Fisher is essentially easy to compute, provided that one is already calculating the gradient
- $\mathrm{rank}(g^x) \leq \mathrm{rank}(g^z) = m$, making optimization strategies for sparse matrices applicable (Lanczos method)
- More optimization tricks to accelerate the computing process

$$\eta^T g^x \eta = \mathbb{E}_{r(y|z)}[(\eta^T(\frac{\partial}{\partial x}\log p(y|x)))^2]$$

Additional constraint

An additional constraint is necessary to guarantee the effectiveness of the objective. Let $\mathcal{J}(y, x)$ be the cross entropy loss of the neural network.

$$\max_{\eta} \eta^T g^x \eta \qquad \text{s.t. } \|\eta\|_2^2 = \epsilon, \mathcal{J}(y, x) \leq \mathcal{J}(y, x + \eta)$$

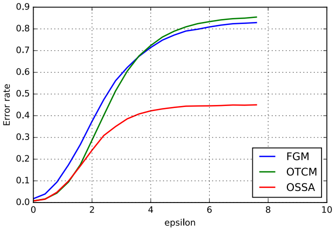
Why is it necessary?

- Let $\tilde{\eta} = -\eta$ be the opposite-direction-perturbation.

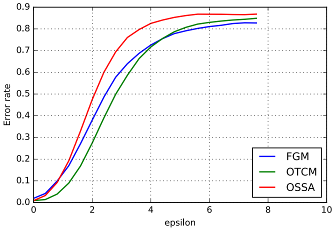
$$\eta^T g^x \eta = \tilde{\eta}^T g^x \tilde{\eta}$$

- but the two directions are not equivalent

Additional constraint

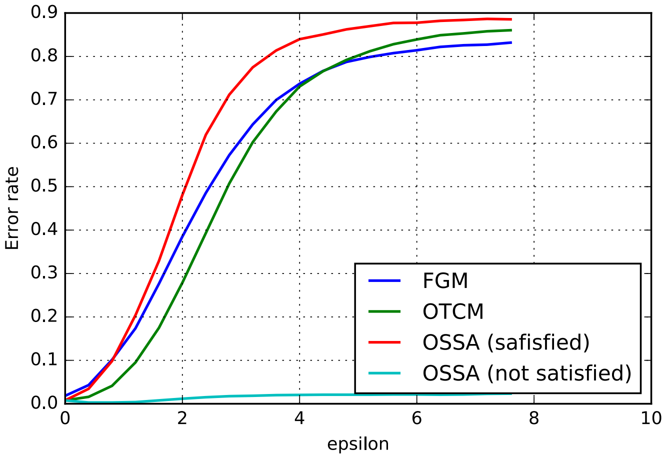


(a) without additional constraint



(b) with the constraint

Additional constraints



Additional constraint

- The empirical distribution $r(y|z)$ is written as

$$r(y|z) = \prod_{i=0}^{n} z_i^{y_i}$$

Additional constraint

- The empirical distribution $r(y|z)$ is written as

$$r(y|z) = \prod_{i=0}^{n} z_i^{y_i}$$

- This makes FIM a diagonal matrix

$$g_{ij}^{z} = \begin{cases} \frac{1}{z_i} & \text{if} \quad i = j \\ 0 & otherwise \end{cases}$$

Additional constraint

- The empirical distribution $r(y|z)$ is written as

$$r(y|z) = \prod_{i=0}^{n} z_i^{y_i}$$

- This makes FIM a diagonal matrix
- Since the network is piecewise linear, we can assume constant Jacobian field in a neighborhood, then

$$\int_0^{\epsilon} \sqrt{\dot{\eta}_i \dot{\eta}_j g_{ij}^x}\, ds \geq \int_0^{\epsilon} \sqrt{\dot{\tilde{\eta}}_i \dot{\tilde{\eta}}_j g_{ij}^x}\, ds$$

Fisher information matrix on large datasets

The outer product is an inefficient representation

- Observe that $\eta^T g^x \eta = \mathbb{E}_{r(y|z)}[(\eta^T \frac{\partial}{\partial x} \log p(y|x))^2]$
- Similarly, $g^x \eta = \mathbb{E}_{r(y|x)}[(\eta^T \frac{\partial}{\partial x} \log p(y|x))(\frac{\partial}{\partial x} \log p(y|x))]$

Fisher information matrix on large datasets

Algorithm 1: One Step Spectral Attack (Power iteration)

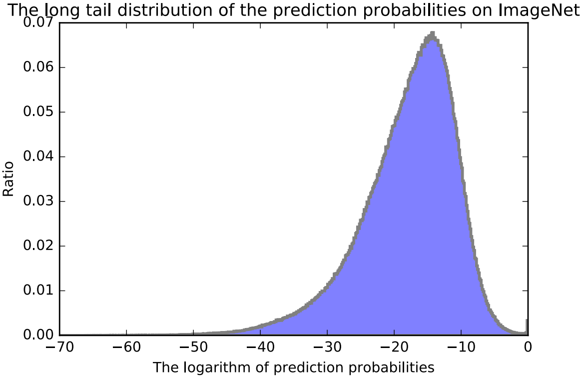
Input: input sample $\mathbf{x}$, corresponding labels y, a deep learning model with the output $p(y|\mathbf{x})$ and the loss $\mathcal{J}(y, \mathbf{x})$.

Output: the perturbation $\boldsymbol{\eta}$, the greatest eigenvalue λ^*.

Initialize $\boldsymbol{\eta}$ as an random vector with unit norm;
while $\boldsymbol{\eta}$ *not converged* **do**
    Update $\boldsymbol{\eta} \leftarrow \mathbb{E}_{y|\mathbf{x}}[((\frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}))^T \boldsymbol{\eta})(\frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}))]$;
    Normalize $\boldsymbol{\eta} \leftarrow \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2}$;
end
if $\mathcal{J}(\mathbf{x} + \boldsymbol{\eta}) \leq \mathcal{J}(\mathbf{x})$ **then**
    $\boldsymbol{\eta} \leftarrow -\boldsymbol{\eta}$;
end

Fisher information matrix on large datasets

For datasets with a large number of categories (e.g. ImageNet), the expectation can also be time consuming.



Fisher information matrix on large datasets

- **Solution**: Monte Carlo sampling from $r(y|z)$.
- Empirically, we found $\frac{1}{5}$ iterations of sampling is good enough for the approximation.
- In practice, we adopt the alias method to perform the sampling from $r(y|z)$ with $O(1)$ time complexity[10].

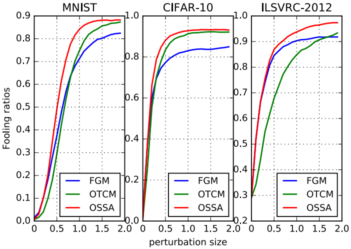
[10] G. Marsaglia, W. W. Tsang, and J. Wang. "Fast generation of discrete random variables". In: *Journal of Statistical Software* 11.3 (2004), pp. 17–24.

Fisher information matrix on large datasets

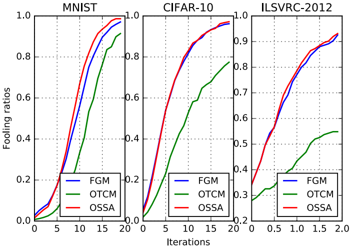
- What if we want a group of orthonormal basis representing the space of adversarial examples?
- **Solution**: Lanczos algorithm[11]
- Particularly efficient for sparse matrices, yielding a total time complexity of $O(dmn)$

[11] D. Calvetti, L. Reichel, and D. C. Sorensen. "An implicit restarted Lanczos method for large symmetric eigenvalue problems". In: *Electronic Transactions on Numerical Analysis* 2 (1994), pp. 1–21.

Fisher information matrix on large datasets



(c) One-step attack

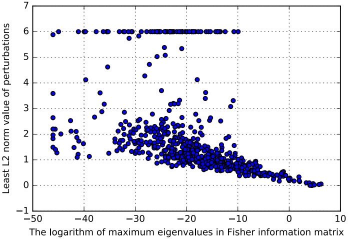


(d) Iterative attack

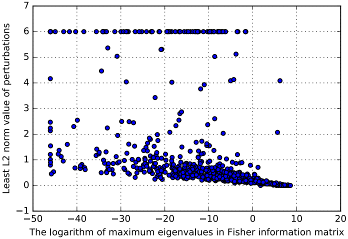
Outline

Empirical evidence

Visualizing the vulnerability measured by the eigenvalues of FIM



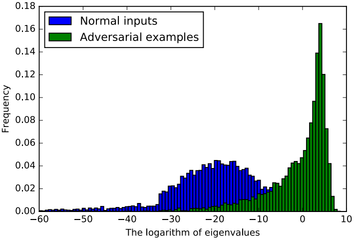
(e) MNIST



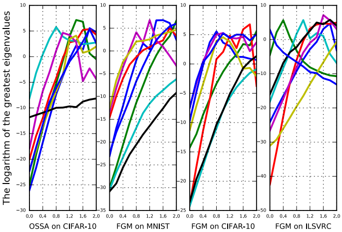
(f) CIFAR-10

Empirical evidence

Why is it practical to distinguish the adversarial examples via the eigenvalues of Fisher information matrix?



(g) statistical histogram of Fisher information matrix eigenvalues



(h) increasing of eigenvalues along the perturbation direction

Why exponential?

Observe that the eigenvalues increases exponentially with the linear decreasing of the least ℓ_2 adversarial perturbation size.
Several pieces of the jigsaw puzzle including:

- The quadratic form $\eta^T g^x \eta = \eta^T J_f^T g^z J_f \eta$ is an approximation of the Fisher information metric on the tangent space of a given sample x
- There exists an exponential mapping $Exp_x(\eta) : T_x\mathcal{M} \to \mathcal{M}$ from the tangent space to the geodesic on $\mathcal{M}$

Why exponential?

Observe that the eigenvalues increases exponentially with the linear decreasing of the least ℓ_2 adversarial perturbation size.
Several pieces of the jigsaw puzzle including:

- Geodesic distance

$$\int_0^{\epsilon} \sqrt{\dot{\eta}_i \dot{\eta}_j g_{ij}^x} ds = \sqrt{8\mathrm{JSD}(p(y|x)||p(y|x+\eta))}$$

- Jensen Shannon divergence is a bounded measure for pdfs
- When η is optimal, the greatest eigenvalue $\|\eta\|^2 e^* = \eta^T g^x \eta$
- ...

Adversarial detection

Key idea: using an auxiliary classifier to distinguish the adversarial examples with the eigenvalues of FIM serving as characteristics.
Other practical techniques

- We use the logarithm of the eigenvalues as the features for classification
- The aforementioned Lanczos algorithm is adopted to calculate a group of orthonormal basis
- The positive set of the training set is composed of both normal samples and noisy samples[12]

[12] A. Fawzi, M. Seyed D. Moosavi, and P. Frossard. "Robustness of classifiers: From adversarial to random noise". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1632–1640.

Evaluations

Table: The AUC scores of detecting adversarial attacks using random forest classifiers and eigenvalues of FIM as characteristics

	MNIST				
AUC (%)	FGM	OTCM	Opt	BIM	OSSA
KD	78.12	95.46	95.15	98.61	84.24
BU	32.37	91.55	71.30	25.46	74.21
KD+BU	82.43	95.78	95.35	98.81	85.97
Ours	**96.11**	**98.47**	**95.67**	**99.10**	**93.13**

Evaluations

Table: The AUC scores of detecting adversarial attacks using random forest classifiers and eigenvalues of FIM as characteristics

	CIFAR-10				
AUC (%)	FGM	OTCM	Opt	BIM	OSSA
KD	64.92	92.13	91.35	98.70	88.89
BU	70.40	91.93	91.39	97.32	87.44
KD+BU	76.40	**94.45**	93.77	98.90	93.54
Ours	**80.18**	93.68	**99.45**	**99.43**	**98.01**

Generalization ability

Table: The generalization ability for detecting adversarial attacks

AUC (%)	Tested on				
Trained on	FGM	OTCM	Opt	BIM	**OSSA**
FGM	94.31	91.92	90.78	91.87	92.13
OTCM	98.55	98.96	98.26	97.78	98.57
Opt	95.18	95.30	96.90	97.15	96.11
BIM	98.10	96.00	97.09	98.57	96.35
OSSA	91.17	91.47	89.77	89.47	89.67

Bad case analysis

Unfortunately, the defence mechanism is specifically designed under an ℓ_2 norm framework, making it almost completely failed to resolve the ℓ_0 norm cases

AUC (%)			Tested on			
Trained on	FGM	OTCM	Opt	BIM	OSSA	**JSMA**
FGM	94.31	91.92	90.78	91.87	92.13	75.35
OTCM	98.55	98.96	98.26	97.78	98.57	70.12
Opt	95.18	95.30	96.90	97.15	96.11	68.78
BIM	98.10	96.00	97.09	98.57	96.35	57.86
OSSA	91.17	91.47	89.77	89.47	89.67	65.40
JSMA	40.99	58.46	50.11	60.23	50.18	49.88

Summary

- Compared with KD, better at distinguishing iterative attacks and adversarial perturbations obtained via binary search

Intuitively, for a binary softmax classifier without hidden layer, i.e. $z_i = w_i x$ and $y_i = \frac{\exp z_i}{\sum_i \exp z_i}$, $i = 1, 2$, the eigenvalue is

$$
\begin{aligned}
e^* &= y_1 (\frac{\partial \log y_1}{\partial x})^2 + y_2 (\frac{\partial \log y_2}{\partial x})^2 \\
&= (w_1 - w_2)^2 y_1 y_2,
\end{aligned}
$$

where $y_1 y_2$ is actually a quadratic function taking maximum value in $y_1 = y_2 = 0.5$.

Some computational tricks used here can not be extended to high dimensional space (please notify me if I'm wrong)

Summary

- Compared with KD, better at distinguishing iterative attacks and adversarial perturbations obtained via binary search
- Using the methods described in[13] to bypass our method would be extremely inefficient (almost intractable for large datasets)

$$\ell_{\text{total}}(\eta) = ||\eta||^2 + \alpha\ell(x+\eta) + \beta(\eta^T g^{x'} \eta),$$

where $x' = x + \eta$.

[13] N. Carlini. and D. Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *ArXiv preprint arXiv: 1705.07263* (2017).

Summary

- Compared with KD, better at distinguishing iterative attacks and adversarial perturbations obtained via binary search
- Using the methods described in[13] to bypass our method would be extremely inefficient (almost intractable for large datasets)
- Our method does not require batch input

[13] N. Carlini. and D. Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *ArXiv preprint arXiv: 1705.07263* (2017).

Thanks!

51174506043@stu.ecnu.edu.cn